

IN THE CLAIMS:

Please amend the claims as follows:

1. (Currently Amended) A method of optimizing the execution of queries in a logically partitioned system, comprising:
monitoring queries executed in a primary processing environment, wherein the primary processing environment comprises a first set of logical partitions of the system wherein each logical partition in the first set of logical partitions is controlled by a respective operating system associated with the logical partition;
selecting a query to be optimized, from the monitored queries;
generating a plurality of access plans for the selected query; and
determining an optimal access plan by executing the query in a background processing environment using the generated access plans, wherein the background processing environment comprises a second set of logical partitions of the system wherein each logical partition in the second set of logical partitions is controlled by a respective operating system associated with the logical partition ; and
in response to determining the optimal access plan, performing subsequent executions of the query using the optimal access plan.
2. (Cancelled) The method of claim 1, wherein the system is a logically partitioned computer system, the primary processing environment comprises one or more primary logical partitions, and the background processing environment comprises one or more background logical partitions.
3. (Original) The method of claim 1, wherein determining an optimal access plan comprises:
repeatedly executing the query in the background processing environment using each of the plurality of access plans, recording a corresponding execution time for each;
and

selecting, as the optimal access plan, an access plan with a lowest corresponding execution time.

4. (Original) The method of claim 1, wherein determining an optimal access plan comprises:

executing the query in the background processing environment using different ones of the plurality of access plans; and

selecting, as the optimal access plan, an access plan with a corresponding execution time below a specified threshold execution time or an access plan with a lowest corresponding execution time.

5. (Original) The method of claim 4, further comprising halting execution of the query in the background processing environment using an access plan if a specified execution timeout period is exceeded.

6. (Original) The method of claim 1, wherein determining an optimal access plan comprises:

performing multiple executions of the query in the background processing environment using the same access plan;

for each of the multiple executions, recording a corresponding execution time; and

selecting, as the optimal access plan, the same access plan only if none of the corresponding execution times exceeds a specified threshold execution time.

7. (Currently Amended) The method of claim 1, wherein:

monitoring queries executed in the primary processing environment comprises storing executed queries in a cache; and

selecting [a] the query to be optimized comprises selecting, from the cache, [a] the query based on the number of times ~~#~~ the query has been executed.

8. (Original) The method of claim 6, wherein selecting a query to be optimized comprises selecting, from the cache, a query having a corresponding execution time above a predetermined threshold.
9. (Original) The method of claim 1, further comprising creating a query execution record for the selected query containing the optimal access plan.
10. (Original) The method of claim 9, wherein the query execution record also contains an execution time corresponding to the optimal access plan.
11. (Original) The method of claim 9, further comprising, subsequent to determining the optimal access plan:
- detecting a request to execute the selected query in the primary partition;
 - obtaining the optimal access plan from the query execution record for the selected query; and
 - executing the selected query in the primary processing environment using the optimal access plan.
12. (Currently Amended) A method of executing queries in a logically partitioned system having at least one primary partition and at least one background partition, the method comprising:
- receiving a query to be executed in the primary partition, wherein the primary partition is controlled by a first operating system;
 - analyzing the query in the background partition to generate one or more performance measures, wherein the background partition is controlled by a second operating system; and
 - executing the query in the primary partition only if the one or more performance measures fall within predefined performance criteria.

13. (Original) The method of claim 12, wherein analyzing the query in the background partition comprises estimating an amount of contention.
14. (Original) The method of claim 13, wherein estimating an amount of contention comprises:
- parsing the query to determine portions of data accessed by the query; and
 - estimating an amount of contention based on stored statistics regarding the portions of data accessed by the query.
15. (Original) The method of claim 12, wherein analyzing the query in the background partition comprises determining an optimal access plan and corresponding execution time for the query.
16. (Original) The method of claim 12, further comprising generating a graphical user interface screen notifying a user if the one or more performance measures do not fall within the predefined performance criteria.
17. (Original) The method of claim 16, wherein the graphical user interface screen provides the user the option of executing the query.
18. (Currently Amended) A computer readable medium containing a program for optimizing query execution which, when executed by a processor, performs operations comprising:
- monitoring queries executed in a primary processing environment, wherein the primary processing environment comprises a first set of logical partitions of a logically partitioned system wherein each logical partition in the first set of logical partitions is controlled by a respective operating system associated with the logical partition;
 - selecting a query to be optimized, from the monitored queries;
 - generating a plurality of access plans for the selected query; and

determining an optimal access plan by executing the query in a background processing environment using the generated access plans, wherein the background processing environment comprises a second set of logical partitions of the system wherein each logical partition in the second set of logical partitions is controlled by a respective operating system associated with the logical partition; and
in response to determining the optimal access plan, performing subsequent executions of the query using the optimal access plan.

19. (Cancelled) The computer readable medium of claim 18, wherein the primary processing environment comprises one or more primary logical partitions and the background processing environment comprises one or more background logical partitions.

20. (Original) The computer readable medium of claim 18, wherein determining an optimal access plan comprises:

repeatedly executing the query in the background processing environment using each of the plurality of access plans, recording a corresponding execution time for each; and

selecting, as the optimal access plan, an access plan with a lowest corresponding execution time.

21. (Original) The computer readable medium of claim 18, wherein determining an optimal access plan comprises:

executing the query in the background processing environment using different ones of the plurality of access plans; and

selecting, as the optimal access plan, an access plan with a corresponding execution time below a specified threshold execution time or an access plan with a lowest corresponding execution time.

22. (Currently Amended) A logically partitioned computer system, comprising:
at least one primary partition, wherein the primary partition is controlled by a first operating system;
at least one background partition, wherein the background partition is controlled by a second operating system;
a first executable component running on the primary partition configured to monitor queries executed in the primary partition, select a query to be optimized, from the monitored queries, and send the selected query to the background partition for determination of an optimal access plan; and
a second executable component running on the background partition configured to receive the selected query, generate a plurality of access plans for the selected query, and determine an optimal access plan by executing the query in the background partition using one or more of the generated access plans,
wherein, after the optimal access plan is determined, the first executable component is configured to perform subsequent executions of the query using the optimal access plan.
23. (Original) The logically partitioned computer system of claim 22, wherein the first executable component is configured to store monitored queries in a query statement cache and select, from the query statement cache, a query to be optimized based on the number of times the query has been executed.
24. (Original) The logically partitioned computer system of claim 22, wherein the second executable component is further configured to create, in a query execution table, a query execution record for the selected query containing the determined optimal access plan.
25. (Original) The logically partitioned computer system of claim 22, wherein the first executable component is further configured to, upon receiving a query to be

executed, retrieve, from the query execution table, a corresponding optimal access plan.